



10653 - Bombs! NO they are Mines

Link submit:

https://uva.onlinejudge.org/index.php?option=onlinejudge&page=show_problem&problem=1594

Solution:

C++: <http://ideone.com/lKvP7s>

Java: <https://ideone.com/16Hlq7>

Python: <https://ideone.com/luBH8Y>

Tóm tắt đề: Cho một mê cung chứa nhiều quả bom được đặt. Nhiệm vụ của bạn là tìm đường đi ngắn nhất từ đỉnh S (bắt đầu), đến đỉnh D(kết thúc). Sao cho không bị dính quả bom nào.

Input

Có nhiều bộ dữ liệu, mỗi bộ test bao gồm các thông tin sau:

- Dòng đầu tiên chứa số dòng R và số cột C của ma trận. $1 \leq R, C \leq 1000$
- Dòng tiếp theo chứa một con số là số lượng dòng có chứa bom $0 \leq \text{rows} \leq R$
- Các dòng tiếp theo lần lượt mô tả mỗi dòng sẽ có bao nhiêu quả bom thông tin bao gồm:
 - Số đầu tiên là index dòng có chứa bom.
 - Số tiếp theo là số lượng bom trên dòng đó.
 - Các số tiếp theo là những quả bom sẽ nằm ở cột nào của dòng đó.
- Dòng tiếp theo, sau mô tả các quả bom, chứa cặp số là tọa độ của đỉnh bắt đầu S.
- Dòng tiếp theo là chứa tọa độ đỉnh cần đến D.

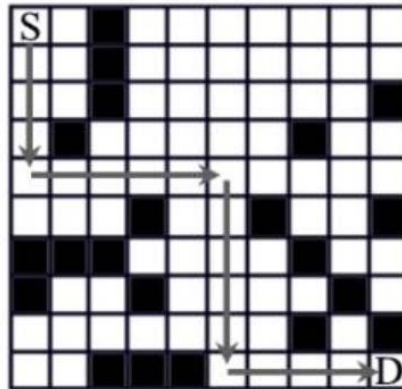
Cuối cùng là 2 số 0 đại diện cho kết thúc bộ test.

Output

In ra một con số duy nhất là số bước đi tối thiểu từ S đến D mà không bị vướng quả bom nào.

10 10 9 0 1 2 1 1 2 2 2 2 9 3 2 1 7 5 3 3 6 9 6 4 0 1 2 7	18
--	----

7 3 0 3 8	
8 2 7 9	
9 3 2 3 4	
0 0	
9 9	
0 0	



Sample input scenario
S- source, D- Destination

Hướng dẫn giải: Ý tưởng để giải quyết bài này tương tự như đi trên mê cung. Vì mỗi bước đi bạn có thể đi 4 hướng: trên, dưới, trái, phải.

Chuyển toàn bộ ma trận vào mê cung maze. Sau đó dùng 2 mảng:

```
int dx[] = { 0, 0, 1, -1 }; int dy[] = { 1, -1, 0, 0 };
```

Để đi các hướng cần thiết trong maze. Mỗi bước đi bạn dùng hàm lưu vết đỉnh cha lại để sau khi tìm được đích đến bạn sẽ theo vết và đếm các bước đã đi được từ S → D.

Độ phức tạp $O(E + V)$.